

University of Groningen

QUANTIFYING THE CONTRIBUTION OF LANGUAGE MODELING TO WRITER-INDEPENDENT ON-LINE HANDWRITING RECOGNITION

Pitrelli, J.F.; Ratzlaf, E.H.

Published in:
EPRINTS-BOOK-TITLE

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2004

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Pitrelli, J. F., & Ratzlaf, E. H. (2004). QUANTIFYING THE CONTRIBUTION OF LANGUAGE MODELING TO WRITER-INDEPENDENT ON-LINE HANDWRITING RECOGNITION. In *EPRINTS-BOOK-TITLE* s.n..

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

QUANTIFYING THE CONTRIBUTION OF LANGUAGE MODELING TO WRITER-INDEPENDENT ON-LINE HANDWRITING RECOGNITION

JOHN F. PITRELLI AND EUGENE H. RATZLAFF

Pen Technologies Group, IBM T. J. Watson Research Center

P. O. Box 218, Yorktown Heights, NY 10598, U. S. A.

E-mail: {pitrelli, ratzlafl}@us.ibm.com

We describe experiments varying the degree of language-model constraint applied to writer-independent on-line handwriting recognition. Six types of models are used, varying statistical components and hard constraints which govern recognition search during the sequencing of characters to form valid texts. Experiments on constrained texts, such as dates and phone numbers, show that although tighter language models cause more inputs to be out-of-domain, they can still eliminate up to 50% of string errors and 75% of character errors compared to using a null language model.

1 Introduction

The IBM on-line handwriting recognition system is an HMM-based system which seeks to label any ink input with the highest-probability word sequence:

$$\hat{w} = \operatorname{argmax}_{w \in \mathcal{W}} P(w|o) \quad (1)$$

where o denotes the observed ink signal, w denotes a word or character sequence, \mathcal{W} denotes the set of all possible such sequences, and \hat{w} denotes the highest-probability sequence. We decompose this probability as follows:

$$P(w|o) = P(o|w)P(w)/P(o) \quad (2)$$

$P(o|w)$ is a statistical model of *how* people write a given text. $P(w)$ is a statistical model of *what* text people are likely to write; we refer to it as the **statistical component of the language model (LM)**. We can decompose w as a succession of words or characters w_1, w_2, \dots :

$$P(w) = P(w_1 w_2 \dots) = P(w_1)P(w_2|w_1)P(w_3|w_1, w_2) \dots \quad (3)$$

with each term of the form $P(w_k|w_1, \dots, w_{k-1})$. These probabilities are often approximated by **N-grams**, $P(w_k|w_{\max(1, k-N+1)}, \dots, w_{k-1})$, clustering histories into equivalence classes based on the most-recent $N-1$ words. This approach is often further simplified by setting $N=1$, resulting in **unigrams**, which are simply one probability per distinct word.

In addition to the statistical component, the LM consists further of a **hard-constraint** component, represented in Equation 1 by $w \in \mathcal{W}$. Hard-constraint components include limited character sets, word sets, and finite-state word grammars. The purposes of hard constraints are to prevent some errors and to save processing time.

The purpose of this study is to quantify the contribution of these language-model components to recognition accuracy. In order to explore the wide range of types of LMs, we focus on various types of constrained texts, such as dates and phone numbers; however, for comparison, an analysis incorporating general-text recognition is included.

2 Background

While the literature contains many investigations of the relationship between language modeling and speech recognition accuracy^{1,2,7}, fewer handwriting recognition studies focus on this aspect of improving accuracy. Marti and Bunke⁴ perform off-line recognition using an HMM / Artificial Neural Network system in which word sets constrain the HMM, but word-bigram processing is performed as a post-process. Kim and Govindaraju³ experiment with a dynamic-matching approach to off-line handwriting recognition, and quantify the benefits of bringing a word-set constraint to bear early in the recognition process, vs. as a post-process. Shridhar *et al.*⁶ also evaluate when to bring lexicon constraints to bear on off-line word recognition.

In contrast, this study explores on-line handwriting recognition. A wide span of LMs is compared for effects on accuracy. Further, all LMs in this study are coupled tightly with recognition, rather than applied as a post-process.

3 Recognition Algorithm Overview

Experiments are performed using the IBM on-line handwriting recognizer in the IBM Ink ManagerTM software; most algorithms are described in detail in previous papers.^{5,8} Data are collected as a stream of (x, y) points indexed in time, re-sampled to be equi-distant. Features based on distances and angles are computed at each point. Windows of temporally-adjacent points are assembled around window centers, which are typically local extrema in x and y . Point feature vectors are spliced together to form window feature vectors. These are projected onto a lower-dimensional space; the resulting vectors are called frames.

Our system is a hidden-Markov-model (HMM) system, in which each character is represented by a set of four allograph models. Each allograph

Table 1. Language-model components.

	Hard constraint	Statistical component
Characters	Limited character set	–
Sequencing of characters into words	Word set	Character N -grams
Sequencing of words into valid texts	Finite-state word grammar	Word N -grams

model consists of varying-length sequences of states. Mixture-Gaussian models, trained using an EM algorithm, represent the distribution of frame vectors for each state. Beam search, governed by any activated hard-constraint LM components, begins with a forward pass using fast-match Gaussians. Hypothesized words are then re-scored using the statistical component of the LM, and then again using detailed-match models.

All experiments reported here are based on running the system in writer-independent mode. Our full character set for American English consists of 93 characters: 26 lower- and 26 upper-case letters, 10 digits, and 31 punctuation marks and other symbols, approximately those on an American-English computer keyboard. Each experiment uses character models trained on a total of approximately 165,000 words plus 330,000 discrete characters provided by 450 writers other than those whose writing was used in the experiments below. Our **baseline** system employs no constraints or statistics to give preference when hypothesizing sequences of these 93 characters. However, our recognizer offers several optional mechanisms implementing the LM components summarized in Table 1 and described in the following sections.

3.1 *Hard-constraint language-model components*

- **Limited character sets**

We can restrict to a subset of the full character set, simply a hard constraint on the characters considered valid at recognition time. For example, for telephone numbers, we limit the character set to the ten digits plus parentheses and hyphen.

- **Word sets**

When the set of possible words is finite, word sets can be applied to constrain the recognizer to hypothesize only words which are in the set.

- **Finite-state word grammars**

Our tightest-constraint LM component is the finite-state word grammar, which provides hard constraint on the sequencing of words. This is appropriate for text types with well-defined and finite word-sequence patterns. An example is dates, in which some variety of sequences is possible, but in a single date there can be only one month, one day in the month, and one year. Grammars are used to prevent *e.g.* a date from lacking a month, or having repeated or contradictory components.

3.2 *Statistical language-model components*

- **Character N -grams**

When full enumeration of valid words is impossible, such as for proper names, one may still employ statistics on the sequencing of characters. By using character N -grams, we enable the recognizer to exploit knowledge such as that `Sinatra` is far more likely than `Zzzqzkj`, without attempting to create an exhaustive set of names to include `Sinatra` explicitly.

- **Word N -grams**

When word-occurrence statistics are available, it is advantageous to supply them to the recognizer. These are particularly important for large-vocabulary recognition, in which disambiguation within sets of similar words is often aided by highly-skewed occurrence statistics. For example, `dog` and `clog` are often written similarly, yet they have very different frequencies of occurrence. Small-vocabulary tasks can benefit as well; for example, for U. S. state codes, `NY` is written very similarly to `NV` but typically has a much higher probability.

3.3 *Relationships among language-model components*

It should be noted that for hard-constraint components, those which constrain larger units subsume those for smaller units. That is, a word set implies a character set, and a grammar is a transition network of word sets. Statistical components, however, complement the hard-constraint components. In particular, character N -grams complement a character set, and word N -grams complement a word set.

Accordingly, a variety of combinations of LM components are possible. In this study, we explore the following sampling of possible combinations:

- Null language model – baseline system (93 equally-likely characters)

- Limited character set, no statistical component
- Limited character set with character unigrams
- Word set, no statistical component
- Word set with word unigrams
- Grammar, no statistical component
- Grammar with word unigrams

4 Databases

Handwriting data were collected using the CrossPadTM Personal Digital Notepad sold by the A. T. Cross Company and IBM Ink ManagerTM software. Writers use an electronic pen which simultaneously puts real ink on real paper and transmits a signal for the pad to pick up as an electronic copy of the handwriting, to be subsequently uploaded to a PC. The “digital ink” consists of a time-ordered sequence of (x, y) co-ordinates from each pen-down time to each pen-up time; pen pressure is not recorded.

Subjects were instructed to write in their own natural style (whether cursive, printed, or mixed), and to write words in order from left to right.

4.1 *Constrained Text*

Between three and ten out of 12 distinct one-page forms were filled out by 121 writers. Example types of forms include merchandise order forms, a medical form, an investment form, and several surveys. Writing was not scripted; subjects were told to imagine themselves in the appropriate situation and fill the form out, altering personal details for the sake of their privacy, while keeping the text believable. While subjects were told to write in their own style, the forms were chosen to contain a variety of writing-space formats, ranging from character-size boxes, to a succession of baselines, to relatively open fields; these implicitly provided varying constraint on style. Furthermore, four forms instruct the writer to print; two of those specify capital letters, and one of those in turn provides examples of how the letters should be written.

Processing the data began by aligning it a page at a time with the corresponding form template. Then the data were sorted automatically according to the form’s fields, accepting any errors which occurred as a result of writer error, overflow of the writing space, paper moving during form-filling, or other registration discrepancies between the form’s position on the notepad

and where the template calls for it to be. Each field’s handwriting was transcribed by hand. Fields were grouped according to content type elicited on the form, *e.g.* phone number, date, surname, address, etc. These content labels are used to trigger the appropriate LMs at recognition time. In these experiments, we do not partition data according to writing-space format. In total, 367 phone-number fields and 713 date fields are used as test data in these experiments.

4.2 General Text

A set of 113 subjects, 11 of whom also participated in constrained-text data collection, wrote several hours of scripted material, in which a “test” set occurred in the middle. The test set consists of 120 “sentences” of text, actually sentences and phrases, some drawn from meeting notes. Five sentences of stimulus text, averaging 45 characters each, appear on a page, with each sentence followed by two blank lines for the subject to write the sentence. Here, subjects were told to try to write on the lines provided on the paper.

For this experiment, each sentence by each writer was classified as “strictly printed” or “cursive”, with “cursive” including mixtures of cursive handwriting and printing. For each of these two categories, one sentence per writer was chosen at random, to generate the general-text database. The database totals approximately 6200 characters in 173 sentences, fewer than two sentences per writer because some provided only printed or cursive writing but not both.

5 Results

Constrained-text results are presented in Table 2. Focus in this study is on contribution of language modeling to accuracy, rather than the absolute accuracy itself, which is conflated by recognition issues not under study here, such as writing neatness and classifier methods. Therefore, results are presented as error rates relative to the baseline system. For example, a relative error rate of 71.5% for phone numbers using a word set means that the use of a word set eliminated 28.5% of the errors on phone numbers compared to the baseline system. For form-filling, we are most interested in **field-error** rates, that is, whether, given the recognition result, a functionally-correct answer would be recovered. For example, for a phone number, only the digit characters are significant; if a punctuation character is misrecognized, but not as a digit, then it does not generate a field error. Similarly, only the first one to three letters of a month name are needed to identify the month uniquely.

We observe a general trend that stronger language modeling reduces error

Table 2. Relative field error rates, in %, on constrained text.

LM	Content Type	
	Phone Numbers	Dates
None / Baseline system	100	100
Limited character set	74.0	96.9
Limited character set + char. unigrams	78.2	80.7
Word set	71.5	86.8
Word set + word unigrams	65.4	81.2
Grammar	60.4	50.2
Grammar + word unigrams	61.8	49.8

Table 3. Out-of-domain rates, in %, for constrained-text inputs.

Hard-Constraint Level	Content Type	
	Phone Numbers	Dates
Character set	3.5	1.3
Word set	8.2	1.7
Grammar	8.7	8.0

rates, as expected because of the reduction in perplexity. This occurs despite the fact that texts which are out of the domain of the hard-constraint component cannot possibly be recognized, and these out-of-domain rates naturally rise as the model tightens, as shown in Table 3. Out-of-domain inputs result from writer error, writers writing in unanticipated ways, and registration problems as described above.

We note that the error-rate patterns vary for different content types. First, limited-character-set constraints aid phone number recognition more than date recognition. This is a consequence of phone numbers using a more-restricted character set than dates, as we neither modeled nor observed alphabetic versions of phone numbers, while dates sometimes have spelled-out month names. In contrast, the strongest improvements to date recognition occur when word-sequencing constraints are applied. We attribute this in part to several small characters, such as apostrophe, comma and hyphen, affecting the meaning of the text and being difficult to recognize, but following strong sequential constraints. Another strong sequencing constraint is that once one word contains any letters, no other may, because only the month may have letters. To exploit such limitations effectively, grammar constraints

are needed. In contrast, for phone numbers, the content “words” are digit sequences, with relatively little sequencing constraint available, limiting gains to be realized by word-set and grammar constraint.

An exception to the general trend is that character unigrams actually hindered phone number recognition slightly, compared to using a limited character set with no statistical component. Many of the added errors were cases in which a 0 or 1 was lost by a small scoring margin under the unigram condition. We attribute this situation to our use of character statistics for American phone numbers in general, which forbids 0 and 1 in two positions in the phone number, and so gives these digits the lowest overall probability. However, all handwriting data were collected in our local area, with area code 914, and so this number occurred disproportionately frequently, and its 1’s were involved in several of these errors. We conclude that supplementing a hard constraint with a suitable statistical component may help recognition in general, but it will always make some valid inputs harder to recognize correctly. Thus, if the statistics are not well-fitted to the task, the error rate may go up, a significant risk in cases in which occurrence statistics are difficult to estimate, *e. g.* how geographically-wide a distribution of phone numbers will be provided by people visiting one location.

Analyzing general text requires two changes. First, a finite-state word grammar is impractical; we cannot put hard constraint on the sequencing of words. Second, there is no counterpart to field error. Instead, we examine **character error**; we compute simply the number of actual characters divided into the number of those which do not appear in the same order in the recognizer output. This is a reasonable error measure because this recognizer generally has character insertion rates very close to its deletion rates.

The word set and word unigrams used for general-text experiments are based on the most-frequently-occurring 33,000 words in a diverse corpus of 338,000,000 words of American English unrelated to the text in this experiment. This word set covers 96.6% of the words in the general-text test set.

Character-error results are shown in Table 4. For the constrained-text conditions, we observe a similar pattern as for field error. However, the error-rate reductions are more dramatic. We attribute this to the fact that correcting some characters does not always lead to making an entire string correct. In general, the LM components providing the greatest benefit vary among text types. The largest improvement for general text comes from word sets, as even a large word set comprises only a small fraction of possible letter sequences and so it represents a substantial constraint. The main error-rate drop for phone numbers comes from the limit on the character set; beyond that, this text type offers the least character-sequencing constraint, yielding

Table 4. Relative character error rates, in %, on constrained and general text.

LM	Content Type		
	General Text	Phone Numbers	Dates
Baseline	100	100	100
Limited character set	100	34.5	88.4
Limited character set + char. unigrams	84.0	40.9	56.2
Word set	46.5	28.9	51.4
Word set + word unigrams	42.8	31.3	48.1
Grammar	N/A	25.4	35.0
Grammar + word unigrams	N/A	28.4	36.2

the least drop in error rates. Dates have a larger character set, but many are letters appearing in only one or two months only when fully spelled out, so applying character unigrams or a word-set constraint improves accuracy, with a secondary improvement resulting from constraining word sequencing.

6 Conclusions

A variety of hard-constraint and statistical language-model components can be employed to improve recognition accuracy substantially. LMs spanning successively larger linguistic units – character, word, full valid text – provide tighter modeling and accordingly reduced recognition error. However, the degree of reduction is subject to several factors. One is the nature of the material; opportunities to apply LMs vary in type and degree among different types of content. Furthermore, as tighter models are applied, the chances of mismatch between the model and the data grows, limiting gains.

7 Future Work

A logical extension of this work will be to incorporate transition probabilities into the finite-state word grammar, thereby co-ordinating the strongest hard constraint with a statistical component.

While word sets provide a major improvement over character-level LM components, a serious shortcoming is that most general-text tasks are not well-represented by a finite word set; some text will remain unrecognizable due to being out of the word set. One solution will be a recognition mode in which a word set can be employed in parallel with a looser model, such as

character N -grams. In this way, we hope to reap most of the accuracy benefit of using word sets, while enabling recognition of rare words not in the set.

Acknowledgments

We gratefully acknowledge Jayashree Subrahmonia and Michael Perrone for feedback and work on the recognizer, Millie Miladinov for her work preparing the databases, and Bruce Lucas and Burn Lewis for assistance with grammars.

References

1. Bellegarda, J. R., "Large Vocabulary Speech Recognition with Multispan Statistical Language Models", *IEEE Transactions on Speech and Audio Processing*, v. 8, no. 1, January, 2000, pp. 76-84.
2. Chen, S. F. and R. Rosenfeld, "A Survey of Smoothing Techniques for ME Models", *IEEE Transactions on Speech and Audio Processing*, v. 8, no. 1, January, 2000, pp. 37-50.
3. Kim, G., and V. Govindaraju, "A Lexicon Driven Approach to Handwritten Word Recognition for Real-Time Applications", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 19, no. 4, April, 1997, pp. 366-379.
4. Marti, U.-V., and H. Bunke, "Towards General Cursive Script Recognition", in Lee, S.-W., ed., *Advances in Handwriting Recognition* (Singapore: World Scientific, 1999), pp. 203-212.
5. Nathan, K. S., H. S. M. Beigi, J. Subrahmonia, G. J. Clary, and H. Maruyama, "Real-Time On-Line Unconstrained Handwriting Recognition using Statistical Methods", *Proceedings of ICASSP 95: IEEE International Conference on Acoustics, Speech, and Signal Processing*, Detroit, Michigan, May 8-12, 1995, v. 4, pp. 2619-2622.
6. Shridhar, M., G. House, and F. Kimura, "Handwritten Word Recognition Using Lexicon Free and Lexicon Directed Word Recognition Algorithms", *Proceedings of the Fourth International Conference on Document Analysis and Recognition (ICDAR)*, August, 1997, v. 2, pp. 861-865.
7. Siu, M., and M. Ostendorf, "Variable N-Grams and Extensions for Conversational Speech Language Modeling", *IEEE Transactions on Speech and Audio Processing*, v. 8, no. 1, January, 2000, pp. 63-75.
8. Subrahmonia, J., K. Nathan, and M. Perrone, "Writer Dependent Recognition of On-Line Unconstrained Handwriting", *Proceedings of ICASSP 96: IEEE International Conference on Acoustics, Speech, and Signal Processing*, Atlanta, Georgia, May 7-11, 1996, v. 6, pp. 3478-3481.